



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAME OF INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/604,987	06/28/2000	Srivatsan Parthasarathy	MS146910.1	6447

27195 7590 01/02/2003

AMIN & TUROCY, LLP
24TH FLOOR, NATIONAL CITY CENTER
1900 EAST NINTH STREET
CLEVELAND, OH 44114

EXAMINER

VU, TUAN A

ART UNIT

PAPER NUMBER

2124

DATE MAILED: 01/02/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/604,987	PARTHASARATHY ET AL.
Examiner	Art Unit	
Tuan A Vu	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 28 June 2000.
 2a) This action is FINAL. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-35 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1-35 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on 06/28/2000 is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 11) The proposed drawing correction filed on _____ is: a) approved b) disapproved by the Examiner.
 If approved, corrected drawings are required in reply to this Office action.
 12) The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
 * See the attached detailed Office action for a list of the certified copies not received.
 14) Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).
 a) The translation of the foreign language provisional application has been received.
 15) Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) Notice of References Cited (PTO-892)
 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
 3) Information Disclosure Statement(s) (PTO-1449) Paper No(s) 2.
 4) Interview Summary (PTO-413) Paper No(s) _____.
 5) Notice of Informal Patent Application (PTO-152)
 6) Other: _____

DETAILED ACTION

1. This action is responsive to the application filed June 28, 2000 and papers submitted on 9/14/2000.

Claims 1-35 have been submitted for examination.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 1-3, 6, 10-12, 15, and 22-35 are rejected under 35 U.S.C. 102(b) as being anticipated by Evans et al., USPN: 5,805,899 (hereinafter Evans).

As per claim 1, Evans discloses a method for version requirement (integrity) checking operable in the linking of dynamic application executable objects at runtime (Fig. 2d), such method comprising:

providing an assembly with manifest (col. 4, line 60 to col. 5, line 2; *Executable Linking Format* (ELF) Object file, Fig. 5, 10; Mapfile 132, Fig. 2c) that contains a list of versioned objects, i.e. modules as claimed, (e.g. *Relocatable object 118*, *Shared object 114*, Fig. 2c-d; *versioned shared objects*, col. 9, lines 20-22) that make up the assembly (e.g. ELF object file; Figs. 5, 10);

providing a manifest with a hash of the contents of at least one objects of the list of sections/modules (e.g. *Section 506*, Fig. 5 and *Hash 614*, Fig. 6).

As per claim 2, Evans further discloses providing the manifest (listing in ELF file) with a hash of each section (module) of the list of sections/modules that constitute the ELF file/assembly (*Version Definition Section 506*, Fig. 5,6 and *versioned shared objects*, col. 8, lines 66 to col. 9, line 7; Fig. 16).

As per claim 3, Evans further discloses providing identity information in the ELF file/assembly (*Section 506*: fields 620, Fig. 6; *Section 510*: fields 1108, 1120, 1122, Fig. 11).

As per claim 6, Evans discloses weak/strong versions of objects to be linked by using the analysis of global variables and pointing to another dependent versions (col. 7, lines 29-56; Fig. 8,15; col. 14, lines 32-41; Fig. 18 – Note: this is equivalent to determining that one version, *weak version*, of objects to assemble is not the latest or has been supplanted by another version, *non-weak version*, i.e. comparing hashes values of the objects contents).

As per claim 10, Evans discloses a method for version compatibility (integrity) checking operable in the linking of dynamic application executable objects, such method comprising:

providing an assembly with manifest (col. 4, line 60 to col. 5, line 2; *Executable Linking Format* (ELF) Object file, Fig. 5, 10; Mapfile 132, Fig. 2c) that contains a list of referenced sections , i.e. assemblies as claimed, (e.g. *section 1...n, dependency section 510* , Fig. 5) that make up the assembly (e.g. ELF object file; Figs. 5, 10);

providing a manifest with a hash of the contents of at least one sections of the list (ELF file) of referenced sections (assemblies) (e.g. *next verdef section → structure #: hash value*, Fig. 16).

As per claim 11, Evans further discloses providing the manifest (listing in ELF file) with a hash (Fig. 10, 11; *section 510*, Fig. 5 and *Hash 1114*, Fig. 11) of each referenced section (assembly) of the list of sections that constitute the ELF file/assembly.

As per claim 12, Evans further discloses providing identity information in the ELF file/assembly (Section 510: fields 1108, 1120, 1122, Fig. 11).

As per claim 15, this is an analogous version to claim 6 in that each limitation 'module' therein is replaced by 'referenced assembly' herein; hence is incorporating the same rejections set forth in claim 6 above.

As per claim 22, Evans discloses a readable medium as addressed in claim 18 above, such medium comprising an assembly (col. 4, line 60 to col. 5, line 2; *Executable Linking Format* (ELF) Object file, Fig. 5, 10; Mapfile 132, Fig. 2c) with a manifest containing a list of referenced sections, i.e. assemblies as claimed (e.g. *section 1...n, dependency section 510* , Fig. 5) that make up the ELF assembly (e.g. ELF object file; Figs. 5, 10) ; a hash of the contents of the manifest of such referenced sections (.g. *next verdef section* → *structure #: hash value*, Fig. 16).

As per claim 23, Evans discloses a system providing a manifest for an assembly (*Executable Linking Format* (ELF) Object file, Fig. 5, 10; Mapfile 132, Fig. 2c), such manifest having a list of versioned objects, i.e. modules as claimed, (e.g. *Relocatable object 118, Shared object 114*, Fig. 2c-d; *versioned shared objects*, col. 9, lines 20-22); providing a manifest with a hash of at least one of the list of versioned objects (e.g. *Section 506*, Fig. 5 and *Hash 614*, Fig. 6).

As per claim 24, Evans discloses components to check versions dependency (e.g. components 126, 122, Fig. 2c-d) accomplishing version checking analogous to that addressed earlier in claim 6 above as far as being compatible with comparing of hash value of modules/versioned objects is concerned. Hence, the rejection in claim 6 herein applies.

As per claim 25, Evans further discloses identity and version information (Section 510: fields 1104, 1108, 1112, 1120, 1122, Fig. 11); a version dependency checker that uses review of originator (e.g. Fig. 15, 16 – Note: a pointer to a versioned object is equivalent to reviewing for correctness (no *Fatal error*) on the structure who points to the referred object) and version information (Fig. 15) and that determine whether 2 object versions are different based on their global variable, and weak flag (col. 7, lines 29-56; Fig. 8,15; col. 14, lines 32-41; Fig. 18 – see claim 6).

As per claim 26, Evans further discloses recording and establishing of version dependency and inheritance (col. 2, lines 17-46; Fig. 2a-b; Fig. 8) and to check runtime version binding (Fig. 2c-d; Fig. 15) based on the version dependency records; and this is equivalent to disclosing binding policy.

As per claim 27, this is a system component version of claim 10 with analogous limitations to those therein, hence incorporates the corresponding rejections of claim 10.

As per claim 28, Evans discloses a third component as addressed in claim 24. Furthermore, this claim is having analogous limitations as those in claim 15 concerning discerning the difference between of the referenced assemblies, i.e. comparing hashes thereof; hence incorporate the rejections in claim 15, in the light of claim 6.

As per claim 29, this is analogous version of claim 25 in the context that the binding policy herein applies to referenced objects; hence incorporates the same rejections of claim 25.

As per claim 30, Evans discloses a system for version requirement (integrity) checking operable in the linking of dynamic application executable objects at runtime (Fig. 2d), such system comprising means for relating a manifest (*Executable Linking Format* (ELF) Object file, Fig. 5, 10; Mapfile 132, Fig. 2c) having a list of at least one related section (e.g. *section 1...n, dependency section 510*, Fig. 5), i.e. assembly as claimed; means for providing the manifest with a hash of that related section (e.g. *Section 506*, Fig. 5 and *Hash 614*, Fig. 6; *next verdef section → structure #: hash value*, Fig. 16).

As per claims 31 and 32, Evans further discloses that one related section is a versioned object, i.e. module (re claim 31) composing a program application executable (e.g. *Relocatable object 118, Shared object 114*, Fig. 2c-d; *versioned shared objects*, col. 9, lines 20-22); that one related section is (re claim 32) being a referenced section (e.g. *section 1...n, dependency section 510*, Fig. 5).

As per claim 33, Evans further discloses means to discern the difference between related sections and/or versioned objects (col. 7, lines 29-56; Fig. 8,15; col. 14, lines 32-41; Fig. 18). Refer to claim 6 for addressing the limitation about comparing hashes.

As per claim 34, Evans further discloses a binding policy, a limitation already set forth in claim 26 and addressed therein.

As per claim 35, Evans further discloses that the related assembly or object is a dynamically linked library (*libc* --col. 4, lines 20-29; *dynamic executable 120*, Fig. 2d).

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 4, 5, 7-9, 13, 14, and 16-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Evans et al., USPN: 5,805,899, in view of Rohatgi et al., USPN: 5,625,693 (hereinafter Rohatgi).

As per claim 4, in reference to claim 3, Evans discloses identity information including version information (e.g. Fig. 5, 6) in the ELF file(assembly of shared versioned objects) or section contents of that file, but **fails to disclose** the providing of the publisher information. Rohatgi, in a program modules linking process analogous to that of Evans, discloses the identity information on the provider (publisher) of the versioned modules (Provider Certificate Descriptor, Provider certificate offset, Fig. 12). It would have been obvious for one of ordinary skill in the art at the time the invention was made to include a publisher or originator of the program components to assemble as taught by Rohatgi to the assembling of versioned objects as disclosed by Evans because this would improve further the version-controlled linking process as taught by Evans as more protection/security (Rohatgi: col. 8, lines 15-36) is applied to the sources providing/publishing those versioned objects.

As per claim 5, in reference to claim 1, Evans discloses hash of the section contents of ELF file/assembly of versioned objects, but **fails to disclose** providing a hash of the contents of such assembly. Rohatgi, in a program modules linking process analogous to that of Evans,

discloses providing a hash of the module Directory (used to gather all linkable modules) contents in a private key (Fig. 12; col. 8, lines 54-60). It would have been obvious for one of ordinary skill in the art at the time the invention was made to include hashing of the assembly which lists program components as taught by Rohatgi to the process of assembling versioned objects as disclosed by Evans because this would impart additional trustworthiness checking as mentioned by Rohatgi (col. 8, line 61 to col. 9, line 7) to the version checking process as taught by Evans as more security is applied to the each instance containing/assembling those versioned objects.

As per claims 7 and 8, in reference to claim 6, Evans discloses the checking of dependency of versioned objects to link (e.g. checking version information in the manifest – instant claim 8) and thereby determining which version is the latest update (Fig. 8, 15) but **fails to disclose** if the publisher of the assembly is (re claim 7) trustworthy, **nor** does Evans disclose checking publisher name information . In view of Rohatgi's teachings about the provider (publisher) of modules applicable to the linking process (see claim 4 above), it would have been obvious for one of ordinary skill in the art at the time the invention was made to include the trustworthiness checking of the publisher as disclosed by Rohatgi as shown in claim 4 above to Evans's method of linking the most updated versions of objects for the same reasons as mentioned in claim 4.

As per claim 9, in reference to claim 1, Evans discloses the pointing within one assembly, of one hash of object to another hash of another versioned object (Fig. 18) **but does not disclose** a manifest providing hash of another manifest upon which it depends on. . In view of Rohatgi's teaching about encrypting in a key (hashing) of the modules directory (assembly) applicable to the linking process (see claim 5 above), it would have been obvious for one of

Art Unit: 2124

ordinary skill in the art at the time the invention was made to include hash the whole assembly of objects (or ELF file contents as taught by Evans) and make it part of the dependency linking scheme as disclosed by Evans. One of ordinary skill in the art would be motivated to do so because this would extend the security/version checking adopted by Evans on objects to the very container of such versioned objects upon which version dependency applies, a form of trustworthiness checking as has been shown in Rohatgi's teaching (col. 8, line 61 to col. 9, line 7).

As per claims 13 and 14, in reference to respectively, claims 12 and 10, Evans discloses both version information for objects and referenced sections (e.g. Section 510: fields 1108, 1120, 1122, Fig. 11) but fails to disclose publisher information (re claim 13) and hash of the assembly (re claim 14) of referenced sections. In view of Rohatgi's teachings about the provider (publisher) of modules applicable to the linking process (see claim 4 above), and a hash of the module Directory contents (see claim 5), it would have been obvious for one of ordinary skill in the art at the time the invention was made to include the trustworthiness checking of the publisher as disclosed by Rohatgi as shown above to Evans's method of linking the most updated versions of referenced sections for the same reasons as mentioned in claim 4 and 5, respectively.

As per claims 16 and 17, in reference to claim 15, these claims recite limitations similar to those of claims 7 and 8, respectively, except that each limitation 'module' therein is replaced by 'referenced assembly' herein; hence is incorporating the same rejections set forth respectively in claims 7 and 8 above, because linking to a referenced section (re claim 16,17) of objects in Evans' linking process is equivalent to linking to a versioned object (re claim 7,8).

Art Unit: 2124

As per claim 18, Evans discloses a computer readable medium (col. 2, line 66 to col. 3, line 16) with an executable for a runtime application program, such medium comprising an assembly (col. 4, line 60 to col. 5, line 2; *Executable Linking Format (ELF) Object file*, Fig. 5, 10; *Mapfile* 132, Fig. 2c) including manifest containing a list of versioned objects (e.g. *Relocatable object* 118, *Shared object* 114, Fig. 2c-d; *versioned shared objects*, col. 9, lines 20-22); but **fails to disclose** a hash of one assembly of a list of such objects (modules) in the manifest. Rohatgi, in a program modules linking process analogous to that of Evans, discloses providing a hash of the module Directory (assembly used to gather all linkable modules) contents in a private key (Fig. 12; col. 8, lines 54-60). It would have been obvious for one of ordinary skill in the art at the time the invention was made to include hashing the contents of the container of the list of objects into a secure entity as taught by Rohatgi to the establishing of linking lists of versioned objects taught by Evans because of the same reasons set forth in claim 5 above.

The following claims 19-21 would be obvious for being dependent on claim 18.

As per claim 19, Evans further discloses including a list of referenced sections, i.e.

assemblies as claimed (e.g. *section 1...n, dependency section 510*, Fig. 5), and a hash of a manifest of one referenced such section (.g. *next verdef section → structure #: hash value*, Fig. 16).

As per claim 20, Evans discloses identity and version information (Section 510: fields 1104, 1108, 1112, 1120, 1122, Fig. 11).

As per claim 21, Evans discloses dynamically linked executables, dlls, from shared library (*libc* --col. 4, lines 20-29; *dynamic executable* 120, Fig. 2d).

Conclusion

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Pat No. 6,324,637 to Hamilton, disclosing plurality of hash index for objects loading.
U.S. Pat No. 5,390,247 to Fisher, disclosing signature and hash of originator of traveling data.
U.S. Pat No. 5,692,047 to McManis, disclosing trusted programs v/s non-trusted programs.
U.S. Pat No. 6,263,379 to Atkinson et al., disclosing linking monikers by Microsoft Corp.
U.S. Pat No. 6,149,318 to Chase et al., disclosing linking of tables and hash value for each type.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (703)305-7207. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks
Washington, D.C. 20231

or faxed to:

(703) 746-7239, (for formal communications intended for entry)

or: (703) 746-7240 (for informal or draft communications, please label
"PROPOSED" or "DRAFT")

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive,
Arlington, VA. , 22202. 4th Floor(Receptionist).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Application/Control Number: 09/604,987
Art Unit: 2124

Page 12

VAT
December 30, 2002

Kakali Chaki

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100